



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Casabona et al.**

Serial No. **09/696,399**

Filed: **October 25, 2000**

For: **Method and System for
Maintaining Software Via Network**

§
§ **Group Art Unit: 2124**
§
§ **Examiner: Vu, Tuan A.**
§
§
§
§

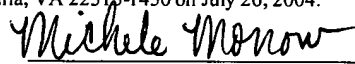
**Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450**

**ATTENTION: Board of Patent Appeals
and Interferences**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on July 26, 2004.

By:


Michele Morrow

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on May 25, 2004.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

07/30/2004 MMEKONEN 00000031 090457 09696399

01 FC:1402 330.00 DA

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interference's that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-16.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: NONE
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1-16
4. Claims allowed: NONE
5. Claims rejected: 1-16

C. CLAIMS ON APPEAL

The claims on appeal are: 1-16.

STATUS OF AMENDMENTS

There are no amendments after final rejection.

SUMMARY OF INVENTION

Summarizing the basic concepts comprised of the present invention, software products are installed and are maintainable in software repositories provided via network on target systems. (Specification, page 17, line 2 to page 18, line 9) This approach resembles a variety of warehouses in which the software products are offered as available goods. (Specification, page 18, line 14 to page 19, line 7) The inventive method and system for software maintenance can be performed in a single command-oriented process. (Specification, page 21, line 16 to page 22, line 10) This centralizes and automates the work that is normally done locally for software maintenance. (Specification, page 25, line 13 to page 26, line 6) In that aspect the installation of fixes, testing and some customization is moved to central organizations, i.e., the repository provided. (Specification, page 31, line 1 to page 32, line 5)

ISSUES

The issues on appeal are whether claims 1, 2 and 4-16 are unpatentable under 35 U.S.C. § 103(a) over Hoyle (U.S. Patent No. 6,141,010) in view of Nguyen et al. (U.S. Patent No. 6,202,070 B1) and whether claim 3 is unpatentable under 35 U.S.C. 103(a) over Hoyle and Nguyen et al., as applied to claim 2, and further in view of Okanoue (U.S. Patent No. 5,689,640).

GROUPING OF CLAIMS

The claims do not stand or fall together. The claims stand or fall in accordance with the following grouping of claims, the reasons set for the following groupings being provided in the following arguments:

- Group I - claims 1, 2, 3, 5, 7, 8, 9, 11-13 and 15;
- Group II - claims 4 and 14; and
- Group III - claims 6, 10 and 16.

ARGUMENT

The Final Office Action rejects claims 1, 2 and 4-16 under 35 U.S.C. § 103(a) as being allegedly unpatentable over Hoyle (U.S. Patent No. 6,141,010) in view of Nguyen et al. (U.S. Patent No. 6,202,070 B1). The Final Office Action rejects claim 3 under 35 U.S.C. 103(a) as being allegedly unpatentable over Hoyle and Nguyen et al., as applied to claim 2, and further in view of Okanoue (U.S. Patent No. 5,689,640). These rejections are respectfully traversed.

Hoyle teaches a method and apparatus for providing an automatically upgradeable software application that displays targeted advertising based upon demographics and user interaction with the computer.

Nguyen teaches a system for software distribution in computer manufacturing which manages and distributes software from release by a software engineering group to installation at a remote manufacturing site or testing facility.

Appellants respectfully submit that, contrary to the allegations made in the Final Office Action, the combination of Hoyle and Nguyen does not, in fact, teach or suggest performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system, as recited in independent claims 1, 7 and 12.

I. 35 U.S.C. § 103, Alleged Obviousness of Groups I-III, Claims 1-16

Claim 1, which is representative of the other rejected independent claims 7 and 12 with respect to similarly recited subject matter, reads as follows:

1. A method for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution wherein the client computer systems are connectable with the at least one central software maintenance institution via a network, the method comprising the steps of:
providing product information for a product in the network system for making the product information available for said plurality of client systems; and
performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system.

Hoyle and Nguyen, taken alone or in combination, fail to teach or suggest performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system.

Hoyle teaches a single repository of files, updated components 48, for a software product. See Hoyle, col. 8, line 64, to col. 9, line 11. As admitted in the Office Action dated June 19, 2003, Hoyle fails to teach or suggest a series of repositories including at least one repository dedicated for a given client system. Nguyen teaches that each software engineering group's database is merged into a single master database. The master database is then replicated to form duplicate master databases at various locations. Also, software is distributed from these databases to local databases at the computer manufacturing sites and test facilities. The local databases will then be used for actual installation of the software onto personal computers. See Nguyen, col. 4, line 65, to col. 5, line 9. In other words, Nguyen teaches a distributed database management system in which master databases are replicated and ultimately distributed to local databases. However, the

actual installation action comprises downloading software components only from the local database of the manufacturing site.

In contradistinction, the present invention provides a software maintenance action comprising downloading data required for a software maintenance action from a sequence of repositories. According to the claimed invention, the sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product. The local-level repository provides files that are specific for a given client system.

In the Final Office Action dated February 25, 2004, the Examiner relies on Hoyle as teaching the idea of downloading such that it operates from a set of more than one repository and the use of sites to accommodate software for a user's specification through the ADM server and Ad server. While the ADM server and the Ad server may provide advertising selection and distribution, only the ADM server provides the distribution of upgrades to the client software application. See Hoyle col. 8, lines 64-66. Thus, as discussed above, Hoyle teaches a single repository of files, updated components 48, for a software product and does not teach performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories.

Moreover, in the Final Office Action the Examiner relies on Nguyen as teaching a hierarchy of repositories going from the most global to a more specialized. Nguyen teaches that each software engineering group's database is merged into a single master database. The master database is then replicated to form duplicate master databases at various locations. Also, software is distributed from these databases to local databases at the computer manufacturing sites and test facilities. Only the local database will then be used for actual installation of the software onto personal computers. See Nguyen col. 4, line 65 to col. 5, line 9. In other words, Nguyen teaches a distributed database management system in which master databases are replicated and ultimately distributed to local databases. However, the actual installation action comprises downloading software components only from the local database of the manufacturing site. In the Advisory Action dated July 15, 2004, the Examiner states that Nguyen's teachings provide evidence of a sequence of repositories for performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action, in mentioning about the local databases only collect data specific to some SKU of manufacturer

in column 8. In column 8, Nguyen merely describes a transfer of software and database information from the master isolated database to master databases and local servers at computer manufacturing facilities is guided by several processes. The transfer from the master isolated database to each computer manufacturing facility's master server is accomplished by making use of DBMS replication techniques. Thus, as discussed above, Nguyen teaches a distributed database management system in which master databases are replicated and ultimately distributed to local databases. However, the actual software installation action comprises downloading software components only from the local database of the manufacturing site. Thus, neither Nguyen nor Hoyle teaches performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories as both Hoyle and Nguyen teach only a single server maintenance function.

Therefore, Hoyle and Nguyen, taken alone or in combination, fail to teach or suggest the features of “performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system,” as recited in claim 1. Since the references, taken alone or in combination, fail to teach or suggest each and every claim limitation, claim 1 cannot be rendered obvious by the proposed combination of Hoyle and Nguyen. Independent claims 7 and 12 recite subject matter addressed above with respect to claim 1 and are allowable for at least the same reasons addressed above.

Furthermore, there is no teaching or suggestion in either of the references regarding the desirability or even possibility of combining the teachings of Nguyen with the teachings of Hoyle. That is, there is no problem illustrated in Hoyle for which Nguyen teaches the solution, or vice versa. While both Hoyle and Nguyen are directed to computer apparatus, there is nothing in either reference that would suggest to one of ordinary skill in the art that any desired benefit would be obtained from a combination of the computer apparatus of Hoyle with that of Nguyen.

Moreover, the alleged motivation provided in the Final Office Action is based completely on a prior knowledge of Appellants' claimed invention. The Final Office Action alleges that the reason one of ordinary skill in the art would be motivated to combine Hoyle with Nguyen is because “because it would enforce duplicate exclusion from operating higher level or global

software list and more local or machine specific list so as to eliminate unnecessary or redundant item; and also would prevent overhead for reconstructing of installable software list or configuration list at a more specify level of a distribution hierarchy.” This alleged motivation is rooted in the erroneous interpretation of Nguyen as teaching performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories which, as illustrated above, is not actually taught by Nguyen. Thus, the allegation that it would be obvious because it would enforce duplicate exclusion from operating higher level or global software list and more local or machine specific list so as to eliminate unnecessary or redundant item; and also would prevent overhead for reconstructing of installable software list or configuration list at a more specify level of a distribution hierarchy is not based on the teachings of the references but is instead an attempt to recreate Appellants’ claimed invention having first had benefit of Appellants’ disclosure. This is impermissible hindsight reconstruction using Appellants’ disclosure as a guide to make the modifications to the references to arrive at Appellants’ claimed invention and cannot be used as a proper basis for rejecting the pending claims.

More particularly, claims 2, 9 and 13 recite, “wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files.” Neither Hoyle nor Nguyen teaches or suggests the limitations recited in claim 2. Since the references, taken alone or in combination, fail to teach or suggest each and every claim limitation, claim 2 cannot be rendered obvious by the proposed combination of Hoyle and Nguyen.

Claim 3 is rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Hoyle and Nguyen et al., as applied to claim 2, and further in view of Okanoue (U.S. Patent No. 5,689,640). The Office Action dated June 19, 2003 alleges:

As per claim 3, Hoyle does not disclose a fall back to an older program version by inactivate the newer version and activating the older version but teaches download and activation of downloaded components into the application (e.g. col. 14, lines 17-27). The upgrade of a software component followed by it activation and determination as to whether such activation is successful is a well-known concept in software upgrade, as evidence by Okanoue, who discloses, in a network service to update files to a plurality of target nodes, a backup copy of the original file reverted to being active if the downloaded update file fails of to activate successfully (col. 1, line 55 to col. 2, line 4; cutover/rollback -- Fig. 8). It would have been obvious for one of ordinary skill in the art at the time the invention was

made to include the rollback step as suggested by Okanoué to the activation process by Hoyle to use the downloaded files because this would immediately and easily restore the failing system, should it encounters problems in activating the upgrade software file, to its functional state without extraneous clean-up operations or costly operating system complications by reactivating the original backup copy with its inherent machine state.

Claim 3 is allowable at least by virtue of its dependency on claim 2. Okanoué fails to make up for the deficiencies of Hoyle and Nguyen. Therefore, the prior art as a whole fails to teach or suggest each and every claim limitation. As such, claim 3 cannot be rendered obvious by a combination of Hoyle and Okanoué, even as combined with Nguyen. Therefore, the rejection of claim 3 under 35 U.S.C. § 103 is overcome.

Thus, in view of the above, Appellants respectfully submit that the combination of teachings alleged by the Final Office Action does not render the invention recited in independent claims 1, 7 and 12 obvious. Claims 2, 4-6, 8-11 and 13-16 depend from claims 1, 7 and 12, and, thus, are distinguished over the alleged combination of Hoyle and Nguyen for the same reasons as noted above. Accordingly, Appellants respectfully submit that the rejection of claims 1, 2 and 4-16 under 35 U.S.C. § 103(a) should be overturned.

II. 35 U.S.C. § 103, Alleged Obviousness of Group II – Claims 4 and 14

In addition to the above, with regard to claims 4 and 14, Hoyle and Nguyen, taken alone or in combination, fail to teach or suggest “generating an input list of files downloadable from said sequence of repositories, generating a list of files present on said target client system, comparing the list of files downloadable from said sequence of repositories with the list of files present on said target client system, and downloading a plurality of files, wherein the plurality of files includes only files which are not yet present in the target client system.” The Office Action dated June 19, 2003 alleges:

As per claim 4, Hoyle discloses the step of upgrading with generating of an input list downloadable from a server repository (e.g. updated blueprint – col. 13, lines 48-63; step 256 – Fig. 14); generating a list of files present on the target client system and comparing of those lists (e.g. current blue print – col. 20, lines 19-32); and downloading only files which are not yet present in the target system (e.g. col. 20, lines 26-42).

But Hoyle fails to specify that the downloadable input list is retrieved from

at least two repositories. But in view of the combined teachings by Hoyle and Nguyen in addressing the use of a sequence of databases to improve the duplication elimination and overhead resource imparting as set forth in claim 1, this limitation herein would have been obvious for the same rationale as set forth therein.

As characterized in the Office Action, Hoyle only teaches a single “blue print” for downloading files. Nguyen actually teaches away from the present invention, because Nguyen teaches an elimination of duplicates. A cited portion of Nguyen states:

One of the features disclosed in this software distribution system is the use of a series of databases to effect the elimination of duplicates. The system architecture is structured in a manner that eliminates duplicate software releases by the various engineering groups. Each transfer of software from database ingest to distribution at the manufacturing installation site, is utilized as an opportunity to discover and eliminate duplicate software releases.

Nguyen, col. 6, lines 10-18.

Thus, it is a goal of Nguyen for there to be only one release of any software product in the local database of the manufacturer.

In contradistinction, the present invention encourages multiple releases, versions, fixes, and customizations to be present in the various levels of the repository hierarchy. The present invention recited in claim 4 allows a client system to select the appropriate files from each level of the repository hierarchy to achieve a proper software version that is specific to the client system. The references, taken alone or in combination, fail to teach or fairly suggest these limitations; therefore, claims 4 and 14 cannot be rendered obvious by a combination of Hoyle and Nguyen.

III. 35 U.S.C. § 103, Alleged Obviousness of Group III – Claims 6, 10 and 16

In addition to the above, with regard to claims 6, 10 and 16, Hoyle and Nguyen, taken alone or in combination, fail to teach or suggest “integrating files into the target system which have been identified by a look-aside procedure as residing in a neighbor system easier to be accessed by the target system than one of said repositories.” The Office Action dated June 19, 2003 alleges:

As per claim 6, Hoyle does not explicitly disclose a look-aside procedure to access in a neighbor system making it easier for integrating the files in the target system but discloses the local pre-store of component files (storage 30 – Fig. 2; col. 14, lines 59-65) to alleviate unnecessary downloading of files from remote

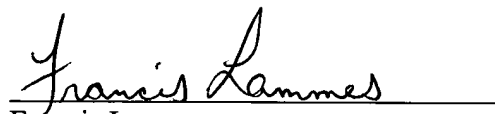
repositories. The look-aside procedure is implied by Hoyle because the technique of storing in the non-remote environment ready files for use in integrating files into the target system is thus equivalent to the technique as to look-aside for the nearest system which would facilitate the retrieval of files as intended for the upgrade because look aside is analogous to not looking further in the remote repositories.

Hoyle only teaches storing files locally on the target system for later use. The claimed feature of the look-aside procedure is not analogous to retrieving files locally. To the contrary, claim 6 expressly recites, “integrating files into the target system which have been identified by a look-aside procedure as residing in a neighbor system.” This limitation is neither taught nor implied in Hoyle. Nguyen fails to provide for the deficiencies of Hoyle. Since the references, taken alone or in combination, fail to teach or suggest each and every claim limitation, claim 6 cannot be rendered obvious by a combination of Hoyle and Nguyen.

CONCLUSION

Based on the foregoing, Appellants respectfully submit that all of claims 1-16 distinguish over the cited art and that the application is in condition for allowance. Accordingly, Appellants respectfully request that the Board of Patent Appeals and Interferences overturn the rejections set forth in the Final Office Action.

Respectfully submitted,

A handwritten signature in black ink that reads "Francis Lammes". The signature is written in a cursive, flowing style. Below the signature is a horizontal line.

Francis Lammes

Reg. No. 55,353

YEE & ASSOCIATES, P.C.

PO Box 802333

Dallas, TX 75380

(972) 367-2001

APPENDIX OF CLAIMS

The text of the claims involved in the appeal are:

1. A method for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution wherein the client computer systems are connectable with the at least one central software maintenance institution via a network, the method comprising the steps of:
 - providing product information for a product in the network system for making the product information available for said plurality of client systems; and
 - performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system.
2. The method according to claim 1 wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files.
3. The method according to claim 2 in which a fall back to an older program version is achieved by inactivating a newer version and activating the older version

4. The method according to claim 2 in which said step of performing said maintenance action serves for an upgrade of a program on at least one target system and said step including the steps of:

generating an input list of files downloadable from said sequence of repositories;

generating a list of files present on said target client system;

comparing the list of files downloadable from said sequence of repositories with the list of files present on said target client system; and

downloading a plurality of files, wherein the plurality of files includes only files which are not yet present in the target client system.

5. The method according to claim 4 in which a total input list is generated by subsequently accessing the repositories and by merging input lists for each repository with a priority of more local files.

6. The method according to claim 1 further comprising the step of integrating files into the target system which have been identified by a look-aside procedure as residing in a neighbor system easier to be accessed by the target system than one of said repositories.

7. A system for maintaining software products, the system comprising:
at least one central software maintenance site;
a network;

a plurality of client computer systems decentralized relative to the at least one central software maintenance site, wherein the client computer systems are connectable with the at least one central software maintenance institution via the network; and

a sequence of repositories, wherein the sequence of repositories provides product information for a product in the network system for making the product information available for said plurality of client systems, wherein said sequence of repositories includes at least a top-level repository storing a complete set of files for the product and a local-level repository storing a first subset of files for the product, wherein the subset of files is specific for a given client system,

wherein a given client computer system from within the plurality of client computer systems performs a software maintenance action for the product by downloading data required for said software maintenance action from the sequence of repositories.

8. The system according to claim 7, wherein the sequence of repositories is provided as a plurality of hierarchically arranged repositories.

9. The system according to claim 7, wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files.

10. The system according to claim 8, further comprising:

at least one neighbor system, wherein the software maintenance action includes integrating files into the target system which have been identified by a look-aside procedure as

residing in the at least one neighbor system easier to be accessed by the target system than one of said repositories.

11. The system according to claim 7, further comprising shadow repositories for at least a subset of the sequence of repositories.

12. A computer program product, in a computer readable medium, for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution wherein the client computer systems are connectable with the at least one central software maintenance institution via a network, the computer program product comprising:

instructions for providing product information for a product in the network system for making the product information available for said plurality of client systems; and

instructions for performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a complete set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system.

13. The computer program product according to claim 12, wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files.

14. The computer program product according to claim 13 in which the instructions for performing said maintenance action serves for an upgrade of a program on at least one target system and the instructions for performing said maintenance action includes:

instructions for generating an input list of files downloadable from said sequence of repositories;

instructions for generating a list of files present on said target client system;

instructions for comparing the list of files downloadable from said sequence of repositories with the list of files present on said target client system; and

instructions for downloading a plurality of files, wherein the plurality of files includes only files which are not yet present in the target client system.

15. The computer program product according to claim 14 in which a total input list is generated by subsequently accessing the repositories and by merging input lists for each repository with a priority of more local files.

16. The computer program product according to claim 12, further comprising instructions for integrating files into the target system which have been identified by a look-aside procedure as residing in a neighbor system easier to be accessed by the target system than one of said repositories.